Monitoring Citrix XenServer

eG Innovations Product Documentation

www.eginnovations.com

Total Performance Visibility

Table of Contents

Table of Figures

Chapter 1: Introduction

Many a times, to obtain an end-to-end view, administrators may seek to monitor performance at the user's desktop. The *Client Desktop* component is used for this purpose. Using this component, administrators can monitor the client desktop in real-time and report on key metrics relating to the health of the desktop such as CPU usage, memory usage, disk activity, paging activity, network traffic, etc. Furthermore, the agent on the *Client Desktop* component includes a software probe that watches all network activity to and from the desktop. By observing all TCP/IP traffic, the eG agent can monitor network latencies and service response times. By comparing the network latencies with service response times, the eG agent is able to differentiate network slowdowns from application slowdowns. Correlation of the desktop resource usage with service performance also allows administrators to clearly identify times when bottlenecks at the client desktops are causing a slowdown in the service performance.

Chapter 1: How to Configure eG Enterprise to Monitor Client Desktop?

An **eg_desktop.ini** file on the agent side drives how the eG agent monitors packets transmissions to and from the client desktop. The example below shows a sample **eg_desktop.ini** file that can be found in the <EG_INSTALL_DIR>\agent\config directory.

```
[EG_CONFIG]

Interface=

Ports=80,1494,7077,53

CacheTime=1

RemoteServers=Web:*:80:C,Dns:*:53:C,Citrix:*:1494:C
```

By default, the eG agent automatically discovers the interface that is to be used for packet capture. By setting the **Interface** value in this file, it is possible to manually override the discovery process. To know what interfaces are available on the system, check the agent log file (<EG_INSTALL_ DIR>\agent\logs\error_log).

The **Ports** specification specifies the ports that the packet capture is set to process. Packets transmitted to other ports are not considered in the traffic analysis done by the eG agent. Note also that the eG agent currently only monitors TCP protocol traffic (i.e., UDP traffic is not analyzed).

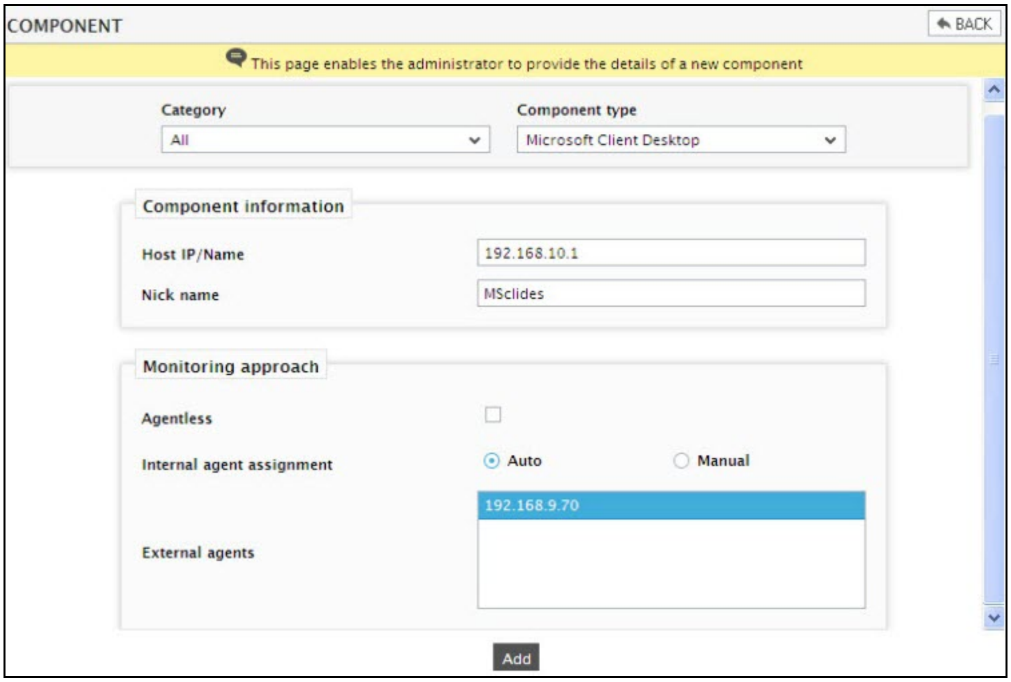
The eG agent can be configured to monitor all traffic on a specific port, or just traffic to specific servers. This configuration is provided in the **RemoteServers** specification. The right hand side setting for this configuration is a comma-separated list. Entries in the list are in the format *name:ip address patterns:portNumber* where the *name* is the display name indicated in the eG monitor interface, the *ip address patterns* is a pattern specifying the IP addresses for which traffic is to be monitored (e.g., *192.168.10.7* specifies a specific server to monitor, while *192.168.10.** represents all servers whose IP addresses match the specified pattern). The port number is the specific port number to be monitored. Multiple entries corresponding to the same name are allowed and for such entries, performance statistics are aggregated while reporting (i.e., *Web:192.168.10.7:80,web:203.197.*:80* is allowed and traffic to all servers matching the IP address pattern will be reported as traffic for the *Web* descriptor).

Once you are done with the configuration steps, manage the *Client Desktop* component using eG Admin interface to monitor the component. The procedure to achieve this is explained in the Section **1.1**.

1.1 Managing the Client Desktop Component

To achieve this, do the following:

1. Log into the eG administrative interface.
2. Add the *Client Desktop* component manually using the **COMPONENTS** page (Infrastructure -> Components -> Add/Modify). Remember that components manually added are managed automatically. Figure 1.1 clearly illustrates the process of adding a *Client Desktop* component.



This page enables the administrator to provide the details of a new component

Category: All

Component type: Microsoft Client Desktop

Component information

Host IP/Name: 192.168.10.1

Nick name: MSclides

Monitoring approach

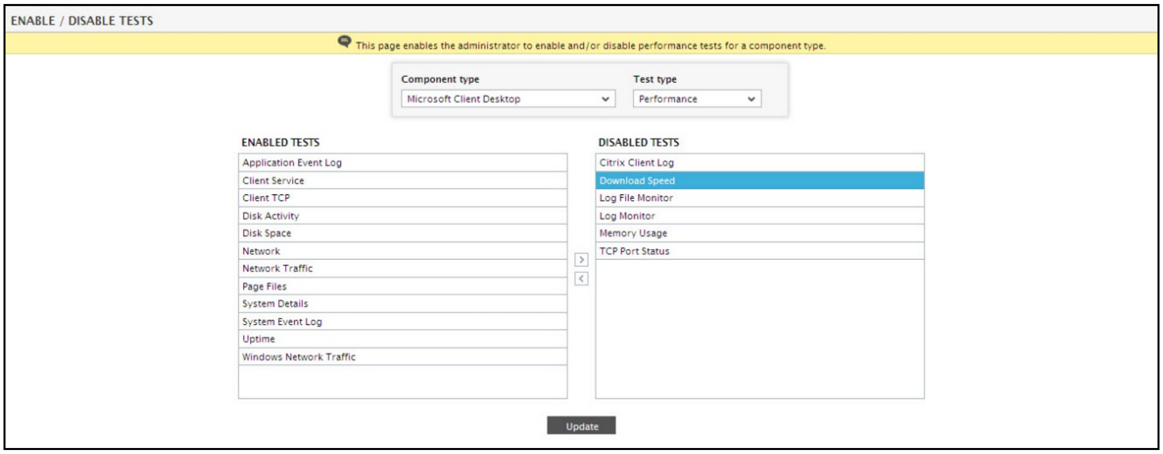
Agentless ☐

Internal agent assignment ● Auto ○ Manual

External agents: 192.168.9.70

Figure 1.1: Adding a Client Desktop component

3. Then, try to sign out of the eG administrative interface. By default the Download Speed test will be disabled. To enable the test, first open the **SPECIFIC TEST CONFIGURATION** page (using the menu sequence: Agents -> Tests -> Specific Configuration), select *Client Desktop* as the component-type, click on the **DISABLED TESTS** link at the top of the list of tests. From the **ENABLE/DISABLE TESTS** page that appears, select the Download Speed test from the **DISABLED TESTS** section and click on the **<<** button. Upon clicking the **Update** button in Figure 1.2, the Download Speed test will be enabled.



ENABLE / DISABLE TESTS

This page enables the administrator to enable and/or disable performance tests for a component type.

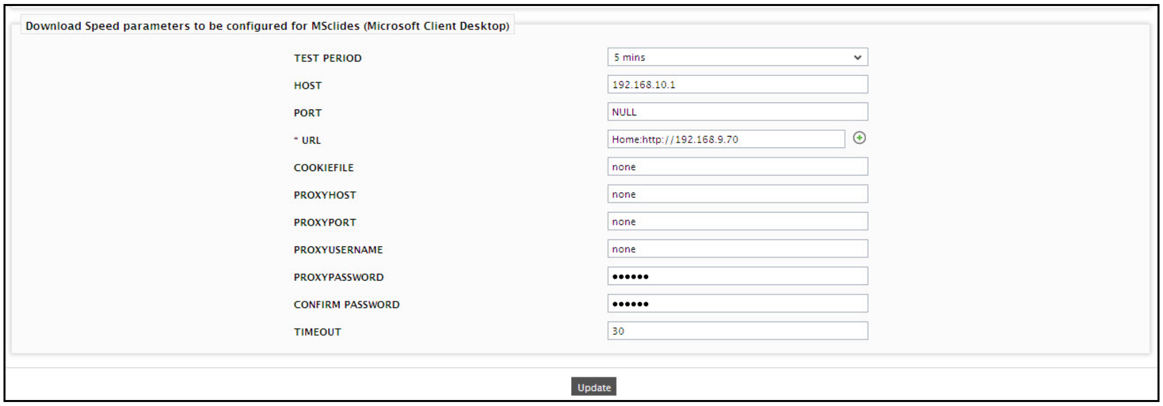
Component type: Microsoft Client Desktop

Test type: Performance

ENABLED TESTS	DISABLED TESTS
Application Event Log	Citrix Client Log
Client Service	Download Speed
Client TCP	Log File Monitor
Disk Activity	Log Monitor
Disk Space	Memory Usage
Network	TCP Port Status
Network Traffic	
Page Files	
System Details	
System Event Log	
Uptime	
Windows Network Traffic	

Figure 1.2: Enabling the Download Speed test for the Client Desktop component

4. Once the test is enabled, you have to manually configure this test. To do so, select the test from the **UNCONFIGURED TESTS** list of the **SPECIFIC TEST CONFIGURATION** page. Figure 1.3 will then appear.Download speed is one of the key indicators of network health. Administrators often download files of varying sizes from sites; a faster download could reduce bandwidth utilization considerably, and save costs. In an era where time is money, slow downloads, can only result in doubling the cost of using a web service. The **DownloadSpeed** test downloads files from a set of configured URLs, and in the process, measures the speed of every file download, thus enabling administrators to accurately judge the efficiency of an internet service and to arrive at service levels.



Download Speed parameters to be configured for MSclides (Microsoft Client Desktop)

TEST PERIOD	5 mins
HOST	192.168.10.1
PORT	NULL
* URL	Home:http://192.168.9.70
COOKIEFILE	none
PROXYHOST	none
PROXYPORT	none
PROXYUSERNAME	none
PROXYPASSWORD	••••••
CONFIRM PASSWORD	••••••
TIMEOUT	30

Figure 1.3: Configuring Download Speed test

5. Finally signout of the eG administrative interface.

Tests mapped to each of the layers of the *Client Desktop* component's layer model (see Figure 1.4), measure the aforesaid activities and report the results of the analysis to the eG manager.

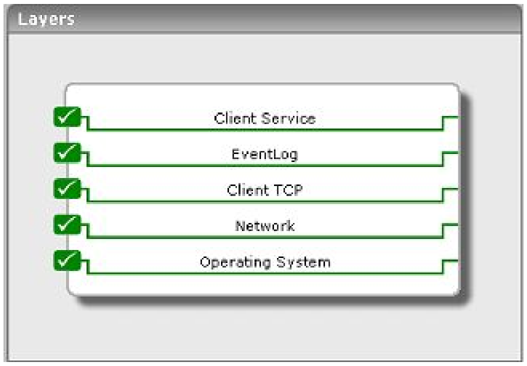


Figure 1.4: The layer model of the Client Desktop component

The **EventLog** layer has been discussed in the *Monitoring Event Logs* document. The **Network** and **Operating System** layers have been dealt with in the *Monitoring Unix and Windows Servers* document. This document therefore will discuss only the **Client TCP** and **Client Service** layers of Figure 1.4.

Note:

Client Desktop requires only a basic monitor license.

1.2 The Client TCP Layer

Using the ClientTcp test, the Client TCP layer measures the TCP traffic to/from the client desktop.

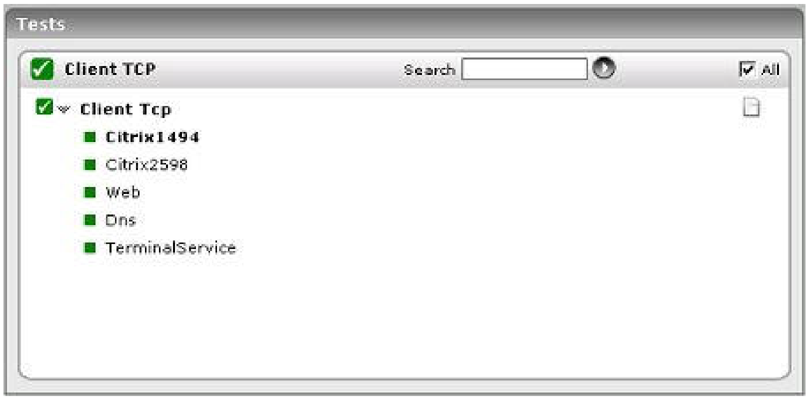


Figure 1.5: The test associated with the Client TCP layer

1.2.1 Client TCP Test

This test reports on the performance of TCP traffic to/from a client desktop. The performance of the TCP layer is impacted significantly by network performance issues - packet loss, congestion, connectivity failures, etc. Hence, by observing the performance at the TCP layer, administrators can easily determine if there is a network issue or not. Since the client desktop may be using different network paths to access different servers, the TCP performance has to be assessed for each server or server group. This test can be executed on Windows boxes only.

Target of the test : A Client Desktop component

Agent deploying the test : An internal agent

Outputs of the test : One set of outputs for every specification against the *RemoteServers* parameter; if the *DynamicServers* specification is uncommented, then one set of results will be reported for every IP address that is being accessed by the client desktop via each of the configured ports.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Adapter Device Selection	By default, the eG agent automatically discovers the interface that is to be used for packet capture. This is why, the Adapter Device Selection flag is set to **Automatic** by default. However, if you want to manually override this discovery process, then, you can do either of the following: • Set the Adapter Device Selection flag to **Manual**, (OR) • Edit the **eg_desktop.ini file** (in the <EG_AGENT_INSTALL_DIR>\agent\config directory) to manually configure the adapter you want the test to use. Both these options have been discussed below: **Setting the Adapter Device Selection flag to Manual** If this is done, then two new parameters, namely - Device Name and Device ID – will automatically appear in the test configuration page. Click on the **Discover** button next to the Device Name parameter to trigger the discovery of the adapters supported by the monitored host. Once discovery is complete, all discovered adapters will populate the Device Name drop-down. From this drop-

Para-meter	Description

down, select the adapter that you want the test to use. As soon as the Device Name is selected, the ID of the chosen adapter will automatically appear against the Device ID box. Then, click on the **Update** button to register the changes.

Editing the *eg_desktop.ini* file to manually specify the adapter name

The **eg_desktop.ini** file on the agent side drives how the eG agent monitors packet transmissions to and from the client desktop. The example below shows a sample **eg_desktop.ini** file that can be found in the <EG_INSTALL_DIR>\agent\config directory.

```
[EG_CONFIG]
Interface=
Ports=80,1494,7077,53,3389,2598
CacheTime=1
RemoteServers=Web:*:80:C,Dns:*:53:C,Citrix1494:*:1494:C,Citrix2598:*:2598:C,TerminalService:*:3389:C
;DynamicServers=80:C,1494:C,2598:C
```

By default, the eG agent automatically discovers the interface that is to be used for packet capture. By setting the **Interface** value in this file, it is possible to manually override the discovery process. To know what interfaces are available on the system, check the agent log file (<EG_INSTALL_DIR>\agent\logs\error_log). For instance, say that the **error_log** of the agent monitoring the client desktop contains the following entries:

```
04/06/2012 06:57:32 INFO Agent: Available packet capture devices are: \Device\NPF_{3B44EC4D-45DB-4276-AC45-00C53206305E}
NOC Extranet Access Adapter (Microsoft's Packet Scheduler) ,\Device\NPF_{F6625292-945E-420B-B207-F4E485BE3625}
DW1530 Wireless- N WLAN Half- Mini Card (Microsoft's Packet Scheduler) \Device\NPF_{DF81BA02-9585-4691-83A5-0420969E0DD9}
Intel(R) 82579LM Gigabit Network Connection (Microsoft's Packet Scheduler)
04/06/2012 06:57:32 INFO Agent: Enabling packet capture enabled using device \Device\NPF_{3B44EC4D-45DB-4276-AC45-00C53206305E}
82579LM Gigabit Network Connection (Microsoft's Packet Scheduler)
```

From these entries, it is evident that the desktop being monitored supports the following adapters:

- \Device\NPF_{3B44EC4D-45DB-4276-AC45-00C53206305E} NOC Extranet Access Adapter (Microsoft's Packet Scheduler)
- \Device\NPF_{F6625292-945E-420B-B207-F4E485BE3625} DW1530 Wireless-N WLAN Half-Mini Card (Microsoft's Packet Scheduler)
- \Device\NPF_{DF81BA02-9585-4691-83A5-0420969E0DD9} Intel(R)

Also, the entry **04/06/2012 06:57:32 INFO Agent: Enabling packet capture enabled using device \Device\NPF_{3B44EC4D-45DB-4276-AC45-00C53206305E}**, clearly indicates that the

Para-meter	Description

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Connection attempts	Indicates the number of TCP connections attempted by the client.	Number	
Connection successes	Indicates the number of TCP connection attempts that succeeded.	Number	
Connection failures	Indicates the number of TCP connection attempts that failed.	Number	Connection failures could be due to performance issues in the interconnection network or at the server end.
Connection status	Indicates the percentage of TCP connection attempts that succeeded.	Percent	A value close to 100 indicates that the network connection is good. A drop in this value is an indicator of poor network or server performance.
Avg. TCP connect time	This measure indicates how long it took on an average to establish a TCP connection.	Secs	When packet loss occurs on the network, TCP uses an exponential back-off algorithm to retry connection establishment. Hence, connection times are likely to grow exponentially as packet loss worsens. A high increase in this metric is an indicator of network connectivity issues (mostly congestion).
Max connect time	This measure indicates the longest TCP connect time during the last measurement period.	Secs	
Out of order transmits	Indicates the number of TCP packets that were received out of order. TCP is a connection-oriented	Number	While out of order transmissions by themselves are not a problem, a large number of our of order transmissions

Measurement	Description	Measurement Unit	Interpretation
	protocol, and in most cases, packets are received in order.		could potentially happen because of packet retransmissions being done at the TCP layer. It is important to monitor retransmissions because TCP throughput and responsiveness decrease drastically with increase in retransmissions. A sudden increase in the out of order transmits or a high percentage of out of order transmissions requires additional investigation. More often than not, such an increase in transmissions is an indicator of a network performance issue.
Percent out of order transmits	The ratio of packets transmitted out of order to packets transmitted. TCP is a connection-oriented protocol, and in most cases, packets are received in order.	Percent	While out of order transmissions by themselves are not a problem, a large number of our of order transmissions could potentially happen because of packet retransmissions being done at the TCP layer. It is important to monitor retransmissions because TCP throughput and responsiveness decrease drastically with increase in retransmissions. A sudden increase in the out of order transmits or a high percentage of out of order transmissions requires additional investigation. More often than not, such an increase in transmissions is an indicator of a network performance issue. A value of 30% or above is a cause for investigation (e.g., use a network sniffer to drill down deeper into the network transmissions).
Out of order packet receptions	Indicates the number of TCP packets received out of order.	Number	Typically, this should be a very low value. A large value is an indicator that potentially a number of

Measurement	Description	Measurement Unit	Interpretation
			retransmissions are happening on the network. Just like packet transmissions (see above), retransmission of packets received can also indicate potential network issues.
Percent out of order packet receptions	Indicates the ratio of packets received out of order to packets received.	Percent	A value greater than 20% requires additional investigation (e.g., use a network sniffer to drill down deeper into the network transmissions).

1.3 The Client Service Layer

The ClientService test associated with this layer monitors performance as seen by the user of a client desktop from a service perspective.

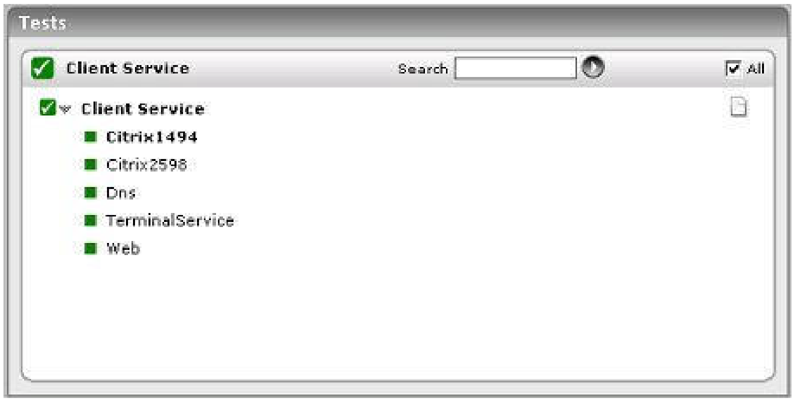


Figure 1.6: The test associated with the Client Service layer

1.3.1 Client Service Test

This test monitors performance as seen by the user of a client desktop from a service perspective. Depending on what servers/ports are configured for monitoring, this test can monitor the performance for user access to Citrix, web, mail and other services. Since this test monitors real user activity from a desktop, rather than simulated activity, the measures of this test are a true reflection of the end user experience. This test can be executed on Windows boxes only.

Target of the test : A Client Desktop component

Agent deploying the test : An internal agent

Outputs of the test : One set of outputs for every specification against the *RemoteServers* parameter; if the *DynamicServers* specification is uncommented, then one set of results will be reported for every IP address that is being accessed by the client desktop via each of the configured ports

Configurable parameters for this test

Parameter	Description
Test Period	How often should the test be executed.
Host	The host for which the test is to be configured.

Adapter Device Selection	By default, the eG agent automatically discovers the interface that is to be used for packet capture. This is why, the Adapter Device Selection flag is set to **Automatic** by default. However, if you want to manually override this discovery process, then, you can do either of the following:
	• Set the Adapter Device Selection flag to **Manual**, (OR)
	• Edit the eg_desktop.ini file (in the <EG_AGENT_INSTALL_DIR>\agent\config directory) to manually configure the adapter you want the test to use.
	Both these options have been discussed below:
	Setting the Adapter Device Selection flag to **Manual**
	If this is done, then two new parameters, namely - device name and device id – will automatically appear in the test configuration page. Click on the **Discover** button next to the device name parameter to trigger the discovery of the adapters supported by the monitored host. Once discovery is complete, all discovered adapters will populate the device name drop-down. From this drop-down, select the adapter that you want test to use. As soon as the device name is selected, the ID of the chosen adapter will automatically appear against the device id box. Finally, click the **Update** button to register the changes.
	Editing the *eg_desktop.ini* file to manually specify the adapter name
	The eg_desktop.ini file on the agent side drives how the eG agent monitors packets transmissions to and from the client desktop. The example below shows a sample *eg_desktop.ini* file that can be found in the <EG_INSTALL_DIR>\agent\config directory.
	`[EG_CONFIG]`
	`Interface=`
	`Ports=80,1494,7077,53,3389,2598`
	`CacheTime=1`

Para-meter	Description

	`RemoteServers=Web:*:80:C,Dns:*:53:C,Citrix1494:*:1494:C,Citrix2598:*:2598:C,TerminalService:*:3389:C`
	`;DynamicServers=80:C,1494:C,2598:C`
	Then, save the file and restart the eG agent.
	Note:
	If you have picked a device name from the admin interface and also manually specified a different device name against the Interface parameter in the *eg_desktop.ini* file, the device name specification will override the Interface specification.
	In addition to specifying the Interface to use, you can also specify the **Ports**, **Cache time**, **RemoteServers**, and **DynamicServers** for the test using the eg_desktop.ini file. The **Ports** specification specifies the ports that the packet capture is set to process. Packets transmitted to other ports are not considered in the traffic analysis done by the eG agent. Note also that the eG agent currently only monitors TCP protocol traffic (i.e., UDP traffic is not analyzed).
	The eG agent can be configured to monitor all traffic on a specific port, or just traffic to specific servers. This configuration is provided in the **RemoteServers** specification. The right hand side setting for this configuration is a comma-separated list. Entries in the list are in the format *name:ip address patterns:portNumber:C*, where the name is the display name indicated in the eG monitor interface, and the ip address patterns is a pattern specifying the IP addresses for which traffic is to be monitored (e.g., 192.168.10.7 specifies a specific server to monitor, while 192.168.10.* represents all servers whose IP addresses match the specified pattern). The port number is the specific port number to be monitored. Multiple entries corresponding to the same name are allowed and for such entries, performance statistics are aggregated while reporting (i.e., *Web:192.168.10.7:80:C,web:203.197.*:80:C* is allowed and traffic to all servers matching the IP address pattern will be reported as traffic for the Web descriptor).
	If you are not aware of the exact IP addresses or IP address patterns of the servers with which the client desktop communicates, then, you can configure the eG agent to monitor all traffic from the client desktop to a specific set of server ports. To achieve this, simply uncomment the **DynamicServers** specification by removing the ';' that precedes this specification. The server ports that the eG agent will be monitoring are specified on the right hand side of this entry in the format, *portnumber:C*.
	To enable the eG agent to monitor more number of ports, you can append to the comma-separated list of ports available on the right hand side of the **DynamicServers** specification. Then, save the file and restart the eG agent.

Measurements of the test

Measurement	Description	Measurement Unit	Interpretation
Avg response time	Indicates the average time from when a data request is sent by the client to when the server returns a response.	Secs	Comparing this value with the response/connection time at the TCP layer provides an indicator of where the bottleneck is. For example, if the service response time is high but network response is low, this implies that there is a slowdown at the application layers and not in the network.
Max response time	Indicates the maximum response time for requests from the client during the last measurement period.	Secs	
Data packets with no response	Indicates the number of times during the last measurement period when a data request was sent by the client but a corresponding response was not received from the server.	Number	Ideally, this value should be low.
No responses percent	Indicates the ratio of the number of data requests for which no response was received to the total number of data requests sent during the last measurement period.	Percent	Depending on the nature of the service being accessed, this value should be near zero. A high value indicates potentially that the client is not receiving responses from the servers it is connecting to.
Data transmitted	Indicates data transmissions from the client desktop during the last measurement period.	KB/Sec	
Data received	Indicates the data receptions by the client desktop during the last measurement period.	KB/Sec	

Note:

For the **ClientTcp Test** and **ClientService Test** to function smoothly, the eG agent on Windows requires the WinPcap library. WinPcap is the industry-standard tool for link-layer network access in Windows environments: it allows applications to capture and transmit network packets bypassing the protocol stack, and has additional useful features, including kernel-level packet filtering, a network statistics engine and support for remote packet capture. WinPcap consists of a driver, that extends the operating system to provide low-level network access, and a library that is used to easily access the low-level network layers.

To enable the eG agent on Windows to use the WinPcap library, you first need to download the library from the URL: http://www.winpcap.org and then install it on the target Windows host.

1.3.2 Download Speed Test

Download speed is one of the key indicators of network health. Administrators often download files of varying sizes from sites; a faster download could reduce bandwidth utilization considerably, and save costs. In an era where time is money, slow downloads, can only result in doubling the cost of using a web service. The **Download Speed** test downloads files from a set of configured URLs, and in the process, measures the speed of every file download, thus enabling administrators to accurately judge the efficiency of an internet service and to arrive at service levels.

This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick *Client Desktop* as the desired **Component type**, set *Performance* as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **<** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Target of the test : A Client Desktop component

Agent deploying the test : An internal agent

Outputs of the test : One set of outputs for every URL being monitored

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port to which the specified Host listens

Parameter	Description
URL	The web page being accessed. While multiple URLs (separated by commas) can be provided, each URL should be of the format *URL name:URL value*. *URL name* is a unique name assigned to the URL, and the *URL value* is the value of the URL. For example, a URL can be specified as *HomePage:http://192.168.10.12:7077/*, where *HomePage* is the *URL name* and *http://192.168.10.12:7077/* is the *URL value*.
CookieFile	Whether any cookies being returned by the web server need to be saved locally and returned with subsequent requests
ProxyHost	The host on which a web proxy server is running (in case a proxy server is to be used).
ProxyPort	The port number on which the web proxy server is listening.
ProxyUserName	The user name of the proxy server.
ProxyPassword	The password of the proxy server.
Confirm Password	Confirm the password by retyping it here.
Content	Is a set of instruction:value pairs that are used to validate the content being returned by the test. If the Content value is *none:none*, no validation is performed. The number of pairs specified in this text box, must be equal to the number of URLs being monitored. The instruction should be one of *Inc* or *Exc*. Inc tells the test that for the content returned by the web server to be valid, the content must include the specified value (a simple string search is done in this case). An instruction of *Exc* instructs the test that the server's output is valid if it does not contain the specified value. In both cases, the content specification can include wild card patterns. For example, an Inc instruction can be *Inc:*Home page**.
Credentials	The DownSpeedTest supports HTTP authentication. The Credentials parameter is to be set if a specific user name / password has to be specified to login to a page. This parameter is a comma separated list of user name:password pairs, one pair for each URL being monitored. A value of none:none indicates that user authorization is not required. Please be sure to check if your web site requires HTTP authentication while configuring this parameter. HTTP authentication typically involves a separate pop-up window when you try to access the page. Many sites uses HTTP POST for obtaining the user name and password and validating the user login. In such cases, the username and password have to be provided as part of the POST information and NOT as part of the Credentials specification for the DownSpeedTest.
TimeOut	Here, specify the maximum duration (in seconds) for which the test will wait for a response from the server. The default TimeOut period is 30 seconds.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Availability	This measurement indicates whether the server was able to respond successfully to the query made by the test.	Percent	Availability failures could be caused by several factors such as the web server process(es) being down, the web server being misconfigured, a network failure, etc. Temporary unavailability may also occur if the web server is overloaded. Availability is determined based on the response code returned by the server. A response code between 200 to 300 indicates that the server is available.
Response time	This measurement indicates the time taken by the server to respond to the requests it receives.	Secs	Response time being high denotes a problem. Poor response times may be due to the server being overloaded or misconfigured. If the URL accessed involves the generation of dynamic content by the server, backend problems (e.g., an overload at the application server or a database failure) can also result in an increase in response time.
TCP connection availability	This measure indicates whether the test managed to establish a TCP connection to the server.	Percent	Failure to establish a TCP connection may imply that either the web server process is not up, or that the process is not operating correctly. In some cases of extreme overload, the failure to establish a TCP connection may be a transient condition. As the load subsides, the server may start functioning properly again.
TCP connect time	This measure quantifies the time for establishing a TCP connection to the web server host.	Secs	Typically, the TCP connection establishment must be very small (of the order of a few milliseconds). Since TCP connection establishment is handled at the OS-level, rather than by

Measurement	Description	Measurement Unit	Interpretation
			the application, an increase in this value signifies a system-level bottleneck on the host that supports the web server.
Server response time	This measure indicates the time period between when the connection was established and when the server sent back a HTTP response header to the client.	Secs	While the total response time may depend on several factors, the server response time is typically, a very good indicator of a server bottleneck (e.g., because all the available server threads or processes are in use).
Response code	The response code returned by the server for the simulated request	Number	A value between 200 and 300 indicates a good response. A 4xx value indicates a problem with the requested content (eg., page not found). A 5xx value indicates a server error.
Content length	The size of the content returned by the server	Kbytes	Typically the content length returned by the server for a specific URL should be the same across time. Any change in this metric may indicate the need for further investigation on the server side.
Content validity	This measure validates whether the server was successful in executing the request made to it.	Percent	A value of 100% indicates that the content returned by the test is valid. A value of 0% indicates that the content may not be valid. This capability for content validation is especially important for multi-tier web applications. For example, a user may not be able to login to the web site but the server may reply back with a valid HTML page where in the error message, say, "Invalid Login" is reported. In this case, the availability will be 100 % (since we got a valid HTML response). If the test is

Measurement	Description	Measurement Unit	Interpretation
			configured such that the content parameter should exclude the string "Invalid Login," in the above scenario content validity would have a value 0.
Data transfer time	Indicates the time taken for the download to complete.	Secs	A consistent increase in this value could be a cause for concern.
Throughput	Indicates the speed of the download.	Kbps	This value is calculated as a ratio of Content_length and Data_xfer_time. Ideally, this value should be high.

1.3.3 Citrix Client Log Test

This test monitors multiple log files for different patterns.

This test is disabled by default. To enable the test, go to the **ENABLE / DISABLE TESTS** page using the menu sequence : Agents -> Tests -> Enable/Disable, pick *Client Desktop* as the desired **Component type**, set *Performance* as the **Test type**, choose the test from the **DISABLED TESTS** list, and click on the **<** button to move the test to the **ENABLED TESTS** list. Finally, click the **Update** button.

Target of the test : A Client desktop component

Agent deploying the test : An internal agent

Outputs of the test : One set of outputs for every AlertFile and SearchPattern combination.

Configurable parameters for the test

Parameter	Description
Test Period	How often should the test be executed.
Host	The IP address of the host for which this test is to be configured.
Port	The port at which the device listens. By default, this will be *NULL*.
AlertFile	In this text box, specify the path to the log file to be monitored. For instance, your AlertFile specification can be: c*:\Citrix\Application Data\ICAClient\wfcwin32.log.* Multiple log file paths can be provided as a comma-separated list. Also, instead of a specific log file path, the path to the directory containing log files can

Parameter	Description
	be provided - eg., _/user/logs_. This ensures that eG monitors the most recent log files in the specified directory. Specific log file name patterns can also be specified. For example, to monitor the latest log files with names containing the string 'slogs', the parameter specification can be, _/tmp/usr/*slogs*_. Here, '*' indicates leading/trailing spaces (as the case may be). In this case, the eG agent first enumerates all the log files in the specified path that match the given pattern, and then picks only the latest log file from the result set for monitoring. You can also configure the path in the following format: _Name@logfilepath_. Here, _Name_ represents the display name of the path being configured. Accordingly, the parameter specification for the 'slogs' example discussed above can be: _slogs@/tmp/usr/*slogs*_. In this case, the display name 'slogs' will alone be displayed as descriptors of the test. Every time this test is executed, the eG agent verifies the following: • Whether any changes have occurred in the size and/or timestamp of the log files that were monitoring during the last measurement period; • Whether any new log files (that match the AlertFile specification) have been newly added since the last measurement period; If a few lines have been added to a log file that was monitored previously, then the eG agent monitors the additions to that log file, and then proceeds to monitor newer log files (if any). If an older log file has been overwritten, then, the eG agent monitors this log file completely, and then proceeds to monitor the newer log files (if any).
SearchPattern	In the SearchPattern text box, enter the specific patterns of alerts to be monitored. The pattern should be in the following format_:<PatternName>:<Pattern>_, where _<PatternName>_ is the pattern name that will be displayed in the monitor interface and _<Pattern>_ is an expression of the form - *expr* or expr or *expr or expr*, etc. A leading '*' signifies any number of leading characters, while a trailing '*' signifies any number of trailing characters. For example, say you specify _CONNECTED:*CONNECTED to*_ in the SearchPattern text box. This indicates that "_CONNECTED_"is the pattern name to be displayed in the monitor interface. "_*CONNECTED to*_" indicates that the test will monitor only those lines in the alert log which embed the phrase "_CONNECTED to_". A single pattern may also be of the form e1+e2, where + signifies an OR condition. That is, the PatternName is matched if either e1 is true or e2 is true. Multiple search patterns can be specified as a comma-separated list. For example: _CONNECTED:*CONNECTED to*,DISCONNECTED:*DISCONNECTED from*_. If the AlertFile specification is of the format _Name@logfilepath_, then the descriptor for

Parameter	Description
	this test in the eG monitor interface will be of the format: *Name:PatternName*. On the other hand, if the AlertFile specification consists only of a comma-separated list of log file paths, then the descriptors will be of the format: *LogFilePath:PatternName*.
Lines	In the Lines text box, specify two numbers in the format x:y. This means that when a line in the alert file matches a particular pattern, then x lines before the matched line and y lines after the matched line will be reported in the detail diagnosis output (in addition to the matched line). The default value here is 0:0. Multiple entries can be provided as a comma-separated list. If you give *1:1* as the value for *Lines*, then this value will be applied to all the patterns specified in the SearchPattern field. If you give *0:0,1:1* as the value for Lines and if the corresponding value in the SearchPattern field is like *CONNECTED:*CONNECTED to*,DISCONNECTED:*DISCONNECTED from** then: *0:0* will be applied to *CONNECTED:*CONNECTED to** pattern *1:1* will be applied to *DISCONNECTED:*DISCONNECTED* from* pattern
Exclude Pattern	Provide a comma-separated list of patterns to be excluded from monitoring in the Exclude Pattern text box. For example *critical*,*exception*. By default, this parameter is set to '*none*'.
UniqueMatch	By default, the UniqueMatch parameter is set to **False**, indicating that, by default, the test checks every line in the log file for the existence of each of the configured SearchPatterns. By setting this parameter to **True**, you can instruct the test to ignore a line and move to the next as soon as a match for one of the configured patterns is found in that line. For example, assume that *Pattern1:*fatal*,Pattern2:*error** is the SearchPattern that has been configured. If UniqueMatch is set to **False**, then the test will read every line in the log file completely to check for the existence of messages embedding the strings 'fatal' and 'error'. If both the patterns are detected in the same line, then the number of matches will be incremented by 2. On the other hand, if UniqueMatch is set to **True**, then the test will read a line only until a match for one of the configured patterns is found and not both. This means that even if the strings 'fatal' and 'error' follow one another in the same line, the test will consider only the first match and not the next. The match count in this case will therefore be incremented by only 1.
RotatingFile	This flag governs the display of descriptors for this test in the eG monitoring console. If this flag is set to **True** and the AlertFile text box contains the full path to a specific (log/text) file, then, the descriptors of this test will be displayed in the following format: *Directory_containing_monitored_file:<SearchPattern>*. For instance, if the AlertFile parameter is set to *c:\eGurkha\logs\syslog.txt*, and RotatingFile is set to **True**, then, your descriptor will be of the following format:

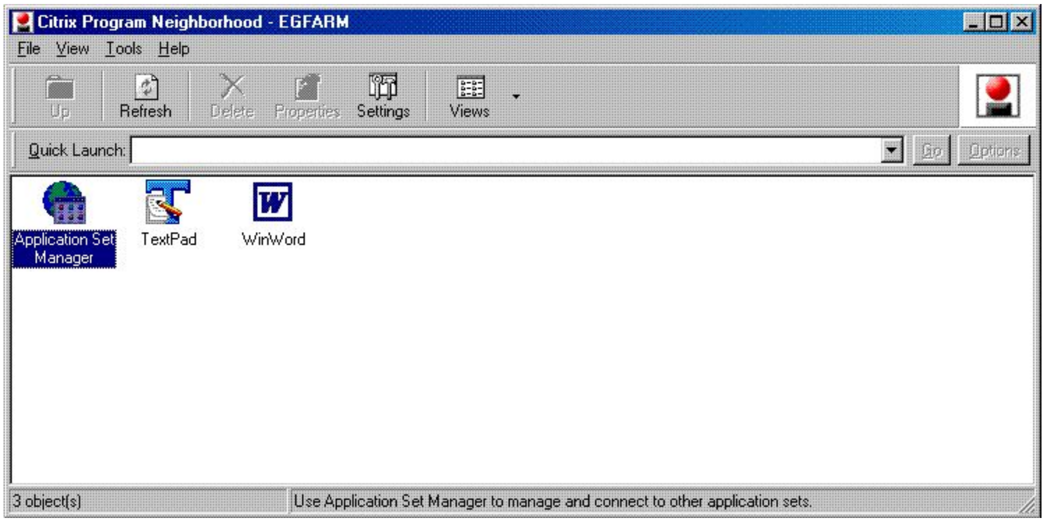
Parameter	Description
	c:\eGurkha\logs:<SearchPattern>. On the other hand, if the RotatingFile flag had been set to **False**, then the descriptors will be of the following format: *<FileName>:<SearchPattern>* - i.e., *syslog.txt:<SearchPattern>* in the case of the example above. If this flag is set to **True** and the alertfile parameter is set to the directory containing log files, then, the descriptors of this test will be displayed in the format: *Configured_directory_path:<SearchPattern>.* For instance, if the AlertFile parameter is set to *c:\eGurkha\logs*, and RotatingFile is set to **True**, then, your descriptor will be: *c:\eGurkha\logs:<SearchPattern>*. On the other hand, if the RotatingFile parameter had been set to **False**, then the descriptors will be of the following format: *Configured_directory:<SearchPattern>* - i.e., *logs:<SearchPattern>* in the case of the example above. If this flag is set to **True** and the alertfile parameter is set to a specific file pattern, then, the descriptors of this test will be of the following format: *<FilePattern>:<SearchPattern>*. For instance, if the AlertFile parameter is set to *c:\eGurkha\logs\*sys**, and RotatingFile is set to **True**, then, your descriptor will be: **sys*:<SearchPattern>*. In this case, the descriptor format will not change even if the RotatingFile flag status is changed .
DD Frequency	Refers to the frequency with which detailed diagnosis measures are to be generated for this test. The default is *1:1*. This indicates that, by default, detailed measures will be generated every time this test runs, and also every time the test detects a problem. You can modify this frequency, if you so desire. Also, if you intend to disable the detailed diagnosis capability for this test, you can do so by specifying *none* against DD Frequency.
Detailed Diagnosis	To make diagnosis more efficient and accurate, the eG Enterprise suite embeds an optional detailed diagnostic capability. With this capability, the eG agents can be configured to run detailed, more elaborate tests as and when specific problems are detected. To enable the detailed diagnosis capability of this test for a particular server, choose the **On** option. To disable the capability, click on the **Off** option. The option to selectively enable/disable the detailed diagnosis capability will be available only if the following conditions are fulfilled: • The eG manager license should allow the detailed diagnosis capability • Both the normal and abnormal frequencies configured for the detailed diagnosis measures should not be 0.

Measurements made by the test

Measurement	Description	Measurement Unit	Interpretation
Recent messages	Indicates the number of messages that were added to the log when the test was last executed.	Number	The value of this measure is a clear indicator of the number of "new" messages that have come into the log of the monitored client desktop.

To set the type of events that need to be logged in the log file, do the following:

1. On a Citrix client install, double-click on the **Citrix Program Neighbourhoold** icon on the desktop.
2. Figure 1.7 will then appear:

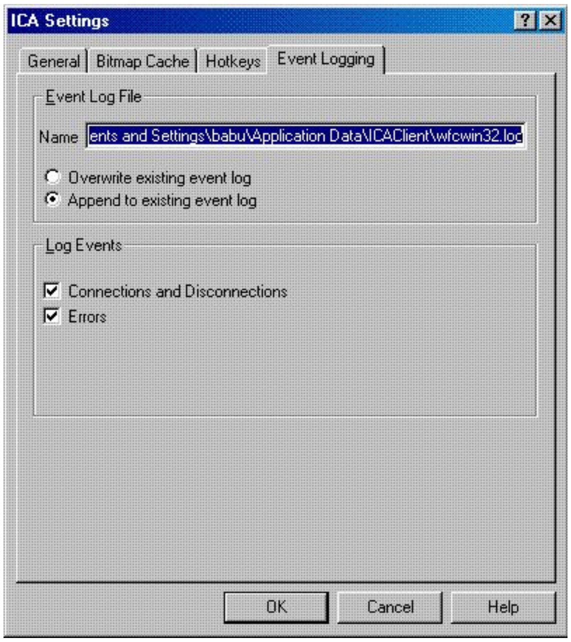


3 object(s)

Use Application Set Manager to manage and connect to other application sets.

Figure 1.7: The Citrix Program Neighbourhood

3. From the **Tools** menu of Figure 1.7, select the **ICA Client** option, and open the **Event Logging** tab page (see Figure 1.8) of the **ICA Settings** dialog box that appears.



Event Log File

Name ents and Settings\babu\Application Data\ICAClient\wfcwin32.log

○ Overwrite existing event log
● Append to existing event log

Log Events

☑ Connections and Disconnections
☑ Errors

Figure 1.8: The ICA Settings dialog box

4. The **Name** text box in Figure 1.8 reveals the complete path to and name of the default event log file. To specify the events that need to be logged in the log file, select either/all of the check boxes in the **Log Events** section of Figure 1.8.

5. Finally, click the **OK** button in Figure 1.8.

1.4 Troubleshooting Client Desktop Monitoring

On some flavors of Windows (particularly, Windows Vista and above), the eG agent monitoring the Client Deskop component may fail to report metrics. Checking the agent error_log may reveal the following error message:

Exception in thread "main" java.lang.UnsatisfiedLinkError: C:\egurkha\lib\Jpcap.dll: Can't find dependent libraries

The desktop agent typically uses a library file named **npptools.dll** to pull out the necessary metrics from the target *Client Desktop*. While Microsoft bundled this dll with older versions of Windows, it does not bundle this dll with relatively newer OS versions such as Windows 7/Vista. The above-mentioned error message is captured by the **error_log** if the *Client Desktop* component being monitored is a Windows operating system into which the **npptools.dll** is not bundled by default. To make sure the desktop agent functions properly on such Windows operating systems as well, you need to copy the **npptools.dll** file (in **C:\Windows\system32**) from older versions of Windows to the **C:\Windows\system32** folder of the target Windows host.

About eG Innovations

eG Innovations provides intelligent performance management solutions that automate and dramatically accelerate the discovery, diagnosis, and resolution of IT performance issues in on-premises, cloud and hybrid environments. Where traditional monitoring tools often fail to provide insight into the performance drivers of business services and user experience, eG Innovations provides total performance visibility across every layer and every tier of the IT infrastructure that supports the business service chain. From desktops to applications, from servers to network and storage, from virtualization to cloud, eG Innovations helps companies proactively discover, instantly diagnose, and rapidly resolve even the most challenging performance and user experience issues.

eG Innovations is dedicated to helping businesses across the globe transform IT service delivery into a competitive advantage and a center for productivity, growth and profit. Many of the world's largest businesses use eG Enterprise to enhance IT service performance, increase operational efficiency, ensure IT effectiveness and deliver on the ROI promise of transformational IT investments across physical, virtual and cloud environments.

To learn more visit www.eginnovations.com.

Contact Us

For support queries, email support@eginnovations.com.

To contact eG Innovations sales team, email sales@eginnovations.com.